

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (previously presented): A method for dynamically determining the health of a service resident on a host machine, comprising:
  - collecting service performance information from the resident service, wherein the collected service information relates to a plurality of performance metrics; and
  - translating the collected service performance information into a generic output relating to current operational performance of the service, wherein the generic output is one of a scriptable interface and an application programming interface, and useable by different performance monitoring tools,
    - wherein the generic output comprises a plurality of service health metrics, and wherein the translating step comprises combining one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.
2. (previously presented): The method of claim 1, wherein the host machine comprises one or more components, further comprising:
  - collecting external performance information from one or more of the one or more components;
  - translating the collected external performance information; and
  - combining the translated external performance information and the translated service performance information to provide the generic output.
3. (cancelled):
4. (original): The method of claim 1, further comprising accessing the generic output to read the health of the service.
5. (original): The method of claim 1, wherein the collecting step comprises reading performance information provided by the service.

6. (original): The method of claim 1, wherein the collecting step comprises deriving performance information from the service.

7. (original): The method of claim 6, wherein the deriving step comprises using a wrapper program to read the performance information.

8. (original): The method of claim 6, wherein the deriving step comprises using a probe program to read the performance information.

9. (cancelled):

10. (cancelled):

11. (currently amended): An apparatus that determines a health of a service resident on a host machine, comprising:

a data collection engine that collects service health information; and

a translation data analysis engine that translates the collected service health information using a health generation algorithm and provides a generic output comprising one or more generic health metrics relating to current operational performance of the service, wherein the generic output is one of a scriptable interface and an application programming interface, and useable by different performance monitoring tools, wherein the collected service health information relates to a plurality of performance metrics, wherein the generic output comprises a plurality of service health metrics, and wherein the translation data analysis engine combines one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

12. (previously presented): The apparatus of claim 11, wherein the host machine comprises one or more external components, wherein the data collection engine collects external performance information from one or more of the one or more external components information, and wherein the translation data analysis engine translates the collected external information using the health generation algorithm to provide the one or more generic health metrics.

13. (cancelled):

14. (previously presented): The apparatus of claim 11, wherein the data collection engine, comprises:

a data query module that reads performance information from the service; and  
a data derivation module that derives performance information from the service.

15. (original): The apparatus of claim 14, wherein the data derivation module derives the performance information from one or more of a wrapper program, a benchmark program and a probe program.

16. (original): The apparatus of claim 11, wherein the health generation algorithm comprises:

a weighting scheme that weights one or more performance information parameters;  
a summation scheme that combines one or more performance information parameters;  
and  
a averaging scheme that averages collected service health information for a service health metric.

17. (original): The apparatus of claim 11, further comprising an interval control engine that receives the service health information at a first time interval and provides an output having a second time interval different from the first time interval.

18. (previously presented): A method for monitoring health data of a service operating on a host machine, comprising:

collecting service performance information from the service;  
collecting external performance information from components of the host machine;  
translating the collected service and external performance information according to a health generation algorithm to generate a generic service health output; and  
providing the generic service health output relating to current operational performance of the service as an output file accessible by performance monitoring tools, wherein the generic service health output is one of a scriptable interface and an application programming interface, and usable by the different performance monitoring tools, wherein the collected

service information relates to a plurality of performance metrics, wherein the generic output comprises a plurality of service health metrics, and wherein the translating step comprises combining one or more of the plurality of performance metrics to provide one or more of the plurality of service health metrics, and wherein the plurality of service health metrics comprises availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

19. (original): The method of claim 18, wherein the step of collecting the service performance information comprises reading first service performance parameters and deriving second service performance parameters, and wherein the step of collecting the external performance information comprises reading first external performance parameters and deriving second external performance parameters.

20. (previously presented): The method of claim 18, further comprising collecting the service performance information on a first time interval and adjusting the first time interval to provide the generic service health output at a second time interval.

21. (previously presented): An apparatus that determines a health of a service, wherein the service operates on a host computer, comprising:

- a collection module that receives performance information related to the service;
- a translation health generator module that applies a rule set to the received performance information and derives generic health metrics therefrom; and
- an output module that outputs the generic health metrics relating to current operational performance of the service, wherein the generic health metrics are in a format usable by different performance monitoring tools, wherein the generic health metrics comprise availability, capacity, throughput, service time, queue length, utilization, service level violations, and user satisfaction.

22. (original): The apparatus of claim 21, wherein the collection module receives external performance information from one or more external services coupled to the host computer and receives internal performance information related to operation of the service on the host computer.

23. (cancelled):

24. (previously presented): The apparatus of claim 21, wherein the generic health metrics is one of a scriptable interface and an application programming interface.